UNITED STATES PATENT APPLICATION

for

A METHOD FOR EFFICIENT USE OF A TRANSMIT ENGINE

Inventors:

Pankaj Chhabra

George Lin

prepared by:

WAGNER, MURABITO & HAO LLP

Two North Market Street

Third Floor

San Jose, CA 95113

(408) 938-9060

A METHOD FOR EFFICIENT USE OF A TRANSMIT ENGINE

FIELD OF INVENTION

5   The present invention relates to the field of computer systems. More specifically, the present invention relates to transmit engines used in transmitting packets of information.

BACKGROUND OF THE INVENTION

10

       A transmit engine is the main engine used by a computer system for transmitting packets. Transmit engines are utilized in several devices, including but not limited to switches, hubs, routers, and network interface cards (NICs). A transmit engine operates by receiving packets issued by software

15   which are then placed into a linked list. The software of the computer system sends the packets to the linked list for transmission. The transmit engine works on the linked list, going from packet to packet down the list, in transmitting the packets appropriately.

20   In a typical working environment, software continuously issues packets to the transmit engine, adding packets to the linked list. Current transmit engine architecture requires that the software stall the transmit engine each time a packet is added to the transmit linked list.

3COM-3478.BCG.US.P/JPW/MJB

The transmit engine must be stalled to avoid corruption of the packet being sent to the linked list and the packets currently on the linked list. At no point does the software know which packet the transmit engine is currently

5   working on the linked list, the transmit engine may be working on any packet. As a result, in order to avoid corruption, the software must first stall the transmit engine, then add the packet to the linked list, and then must unstall the transmit engine.

10   It should also be appreciated that the transmit engine must be stalled before inserting a packet into the linked list to avoid hanging the computer system. If the transmit engine is in the process of sending a packet, and another packet is placed in the linked list, it is possible that the insertion of the new packet will cause the system to hang. As a result, in order to avoid

15   hanging the system, the software must first stall the transmit engine, then add the packet to the linked list, and then must unstall the transmit engine.

The requirement of stalling the transmit engine each time a packet is to be inserted presents problems concerning the speed and efficiency of the

20   transmit engine. Constant interruption of the transmit engine results in the loss of a series of clock cycles. Clock cycles are required for the software to notify the controlling processor to stop the transmit engine, for the processor to notify the transmit engine, for the processor to notify the software that the

transmit engine has stopped, and for the software to insert the new packet. Unstalling the transmit engine results in additional loss of clock cycles.

Accordingly, a need exists for a method for providing efficient use of a
5 transmit engine by not requiring the transmit engine to stall during the adding of a packet to a linked list such that the transmit engine may operate at a faster speed. Furthermore, there is a need for a method for adding packets to a linked list without the risk of corrupting the packet or hanging the hardware. Furthermore, there is a need for such a method that is also commercially
10 economical.

SUMMARY OF THE INVENTION

The present invention provides a method for efficient use of a transmit
engine by not requiring the transmit engine to stall during the adding of a
5   packet to a linked list. The present invention also provides a method for adding
packets to a linked list without the risk of corrupting the packet or hanging the
hardware. The present invention is implemented in a commercially
economical manner. The present invention provides these advantages and
others not specifically mentioned above but described in the sections to follow.

10

A method for providing efficient use of a transmit engine in transmitting
packets and packet directing information is described. The present invention
provides for the efficient use of a transmit engine for transmitting the packets,
sending a request, or performing a direct memory access (DMA) transfer. In
15   one embodiment, two linked lists are used to submit packets and packet
directing information to a transmit engine. When there is no packet to send,
both lists are marked as free. When software desires to send a packet to
hardware, the packet is placed on the first linked list. Software marks the first
linked list as busy. If the transmit engine is not busy with the second linked list,
20   software will start the transmit engine utilizing the first linked list.

While the first linked list is marked as busy, all additional packets are
directed to the second linked list. When the packets of the first linked list have

3COM-3478.BCG.US.P/JPW/MJB

been sent, software marks the first linked list as free. The second linked list is then marked as busy and the transmit engine begins transmitting the packets of the second linked list. While the second linked list is marked as busy, all additional packets are directed to the first linked list.

5

The present invention operates by alternating between the first linked list and the second linked list. One linked list directs the transmit engine and the other linked list receives additional packets without requiring the transmit engine to stall while the packets are received. The present invention provides an efficient transmit engine by not requiring the transmit engine to stop each time a packet is added to a linked list.

10

In one embodiment, the present invention utilizes more than two linked lists. In one embodiment, the transmit engine provides a Quality of Service (QoS) scheme in selecting which linked list to use at any given moment. In one embodiment, certain types of packets can be directed to certain linked lists, where each linked list is given a priority. Only when all linked lists with higher priority are marked as free, would the transmit engine use a linked list of lower priority.

15

20

In another embodiment, the linked lists used are chained together. In the situation where the volume of packets being directed by the transmit engine

is low, linked lists are chained together for a time until the volume of traffic increases.

These and other objects and advantages of the present invention will
5   become obvious to those of ordinary skill in the art after having read the following detailed description of the preferred embodiments which are illustrated in the various drawing figures.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention:

FIGURE 1 illustrates an exemplary computer system platform upon which embodiments of the present invention may be practiced.

FIGURE 2 shows a block diagram of a computer system comprising a transmit engine in accordance with one embodiment of the present invention.

FIGURES 3A and 3B show a flowchart diagram of a process for providing efficient use of a transmit engine in transmitting packets utilizing two linked lists in accordance with one embodiment of the present invention.

FIGURES 4A through 4E show block diagrams illustrating the steps of a process for providing efficient use of a transmit engine in transmitting packets utilizing two linked lists in accordance with one embodiment of the present invention.

FIGURE 5 shows a block diagram of various configurations of memory partitions in accordance with one embodiment of the present invention.

3COM-3478.BCG.US.P/JPW/MJB

DETAILED DESCRIPTION

Refer now to Figure 1 which illustrates an exemplary computer system
100 upon which embodiments of the present invention may be practiced. In
5   general, computer system 100 comprises bus 110 for communicating
information, processor 101 coupled with bus 110 for processing information
and instructions, random access (volatile) memory (RAM) 102 coupled with
bus 110 for storing information and instructions for processor 101, read-only
(non-volatile) memory (ROM) 103 coupled with bus 110 for storing static
10   information and instructions for processor 101, data storage device 104 such
as a magnetic or optical disk and disk drive coupled with bus 110 for storing
information and instructions.

In one embodiment, computer system 100 comprises an optional user
15   output device such as display device 105 coupled to bus 110 for displaying
information to the computer user, an optional user input device such as
alphanumeric input device 106 including alphanumeric and function keys
coupled to bus 110 for communicating information and command selections to
processor 101, and an optional user input device such as cursor control device
20   107 coupled to bus 110 for communicating user input information and
command selections to processor 101. Furthermore, an optional input/output
(I/O) device 108 is used to couple computer system 100 onto, for example, a
network.

3COM-3478.BCG.US.P/JPW/MJB

Display device 105 utilized with computer system 100 may be a liquid

crystal device, cathode ray tube, or other display device suitable for creating

graphic images and alphanumeric characters recognizable to the user. Cursor

5   control device 107 allows the computer user to dynamically signal the two-

dimensional movement of a visible symbol (pointer) on a display screen of

display device 105. Many implementations of the cursor control device are

known in the art including a trackball, mouse, joystick or special keys on

alphanumeric input device 106 capable of signaling movement of a given

10   direction or manner of displacement. It is to be appreciated that the cursor

control 107 also may be directed and/or activated via input from the keyboard

using special keys and key sequence commands. Alternatively, the cursor may

be directed and/or activated via input from a number of specially adapted cursor

directing devices.

15   Figure 2 shows a block diagram of a computer system 200 comprising a

transmit engine in accordance with one embodiment of the present invention.

Hardware 210 and software 220 both run on shared memory 230.

20   In one embodiment, hardware 210 is a transmit engine for transmitting

packets of information. In the present embodiment, software 220 sends

packets to shared memory 230. Shared memory 230 comprises at least one

linked list. A linked list is a list that contains packets intended for transmission

by the transmit engine. Hardware 210 (e.g. a transmit engine) runs on one linked list at a time, going in order in transmitting the packets accordingly, performing on each packet distinctly.

5   In one embodiment, each linked list is a partitioned location a shared memory 230. It should be appreciated that a linked list does not require its own memory, but rather is a virtually partitioned location of one memory (e.g. RAM 102 of Figure 1). In this manner, shared memory 230 can contain multiple linked lists, each being a virtually partitioned location of shared memory 230.

10

       Figure 5 shows a block diagram 500 of various configurations of memory partitions in accordance with one embodiment of the present invention. Memory 510 shows two distinct memories, memory A and memory B. Memory A and memory B each comprise one linked list. Memory 520

15   shows three distinct memories, memory A, memory B and memory C. Each memory comprises one linked list. Memory 530 shows n distinct memories, where n is any number. As above, each memory comprises one linked list.

       Still with reference to Figure 5, memory 540 shows one memory that is

20   virtually partitioned to include n number of memory partitions on one memory. In one embodiment, each virtual partition comprises one linked list. As such, any number of linked lists is operable in the present invention without requiring more than one memory unit.

3COM-3478.BCG.US.P/JPW/MJB

In one embodiment, the present invention is a software invention. The present invention is implemented in software that alters the typical functioning of the software interaction between the software and the transmit engine.

5

Figures 3A and 3B show a flowchart diagram of a process 300 for providing efficient use of a transmit engine in transmitting packets utilizing two linked lists in accordance with one embodiment of the present invention. In one embodiment, process 300 is implemented in software operable to run on

10 a computer system (e.g. computer system 100 of Figure 1).

Referring first to Figure 3A, at step 305, process 300 begins. In one embodiment, when process 300 begins both linked list A and linked list B are marked as "FREE." Marking a list as FREE indicates to the software that

15 packets may be placed on the linked list. Conversely, a linked list is marked as "BUSY" indicates to the software that the transmit engine is busy with the linked list, and packets may not be placed on the linked list.

At step 310, the software places a packet is placed on linked list A, as

20 linked list A is marked FREE. It should be appreciated that packets may be placed on any linked list marked as FREE. The linked lists are interchangeable for purposes of the present inventions. In another

embodiment, a packet is placed on linked list B. Linked list A is used in process 300 for exemplary purposes only.

At step 315, linked list A is marked as BUSY by the software. Upon
5　being marked as BUSY, the transmit engine begins working on the packets of linked list A, transmitting them appropriately.

At step 320, all packets sent by the software to the transmit engine are directed to linked list B. Linked list B can receive packets at the same time the
10　transmit engine is working with linked list A. Placing packets on linked list B does not require that linked list A be stalled. It should be appreciated that linked list A can continue to direct the transmit engine even when linked list B is receiving packets from the software.

15　At step 325, the software receives an interrupt signal from the transmit engine. In one embodiment, linked list A is adapted to provide the software with an interrupt signal upon the completion of transmitting all packets of linked list A. The interrupt signal is placed after the last of the packets located in linked list A.

20

At step 330, linked list A is marked as FREE. In one embodiment, once linked list A has completed transmitting the packets located thereon, linked list A is marked as FREE.

3COM-3478.BCG.US.P/JPW/MJB

At step 335, it is determined whether any packets have been received by linked list B. If no packets have been received, process 300 returns to step 305, where both lists are marked as FREE. If packets have been received by linked list B, process 300 continues on to step 340.

Referring now to Figure 3B, at step 340, linked list B is marked as BUSY by the software. Upon being marked as BUSY, the transmit engine begins working on the packets of linked list B, transmitting them appropriately.

At step 345, all packets sent by the software to the transmit engine are directed to linked list A. Linked list A can receive packets at the same time the transmit engine is working with linked list B. Placing packets on linked list A does not require that linked list B be stalled. It should be appreciated that linked list B can continue to direct the transmit engine even when linked list A is receiving packets from the software.

At step 350, the software receives an interrupt signal from the transmit engine. In one embodiment, linked list B is adapted to provide the software with an interrupt signal upon the completion of transmitting all packets of linked list B. The interrupt signal is placed after the last of the packets located in linked list B.

3COM-3478.BCG.US.P/JPW/MJB

At step 355, linked list B is marked as FREE. In one embodiment, once linked list B has completed transmitting the packets located thereon, linked list B is marked as FREE.

5   At step 360, it is determined whether any packets have been received by linked list A. If no packets have been received, process 300 returns to step 305, where both lists are marked as FREE. If packets have been received by linked list B, process 300 returns to step 315 for directing the packets of linked list A.

10   It should be appreciated that process 300 can be expanded to operate on more than two linked lists. In one embodiment, process 300 is expanded to operated on three linked lists. The present embodiment is desirable, for example, in a situation where the transmit engine operates at a very fast speed. Also, the present embodiment is used where a Quality of Service (QoS) scheme is desired. In a QoS scheme, each linked list is given a priority. Each packet is sent to the appropriate linked list. In one embodiment, the transmit engine only works on a linked list of lower priority when linked lists of higher priority are free.

20

In another embodiment, the linked lists are chained together. The present embodiment is desirable, for example, in a QoS situation where there are multiple linked lists and the number of packets in individual lists are small.

The lists can be combined to save the number of interrupts from the transmit engine to the host computer.

Figures 4A through 4E show block diagrams illustrating the steps of a process for providing efficient use of a transmit engine in transmitting packets utilizing two linked lists in accordance with one embodiment of the present invention.

Referring first to Figure 4A, the initial condition (e.g. step 305 of Figure 3A) is shown. In the present embodiment, both linked list A and linked list B are marked as FREE, and the transmit engine is free because it is not working with either linked list.

Figure 4B illustrates the situation where software places a packet on linked list A for transmission by the transmit engine (e.g. step 310 of Figure 3A). The software marks linked list A as BUSY (e.g. step 315 of Figure 3A).

Figure 4C illustrates the situation where the software desires to send more packets while the transmit engine is working on linked list A. Software places additional packets on linked list B, which is still marked as FREE (e.g. step 320 of Figure 3A). Placing packets on linked list B does not require the software to stall linked list A. The acts of placing packets on linked list B and

the transmit engine working on linked list A operate simultaneously and independently of each other, causing no interference.

Figure 4D illustrates the situation where the transmit engine issues an interrupt signal for linked list A, indicating that the transmit engine has completed all activity with respect to linked list A (e.g. step 325 of Figure 3A). Upon receiving the interrupt signal, the software marks linked list A as FREE (e.g. step 330 of Figure 3A). If there are packets in linked list B, the software marks linked list B as BUSY (e.g. step 340 of Figure 3B).

Figure 4E illustrates the situation where the software desires to send more packets while the transmit engine is working on linked list B. Software places additional packets on linked list A, which is marked as FREE (e.g. step 345 of Figure 3A). The acts of placing packets on linked list A and the transmit engine working on linked list B operate simultaneously and independently of each other, causing no interference.

The present invention is applicable where the software has a series of packets to send to the transmit engine for transmission. The present invention does not require the stalling of the transmit engine to load packets onto the transmit engine, thereby saving clock cycles.

3COM-3478.BCG.US.P/JPW/MJB

The preferred embodiment of the present invention, a method for providing efficient use of a transmit engine in transmitting packet directing information, is thus described. While the present invention has been described in particular embodiments, it should be appreciated that the present invention should not be construed as limited by such embodiments, but rather construed according to the below claims.